Midterm Test #1 الإجابة النموذجية للامتحان الأول

(Sunday 03-11-2013) **Time: 1 1/2 Hrs.**

***Answer All Questions*.**

Q1-(25pts)

a. Write the type of MIPS addressing mode for the following instructions. ***(10pts)***

(i) sw ----------> Base addressing.
(ii) and ----------> Register addressing.
(iii) beq ----------> PC-Relative addressing.
(iv) jal ----------> Pseudo-direct addressing.
(v) sltiu ----------> Immediate addressing.

b. Complete the gaps for each instruction and its machine code in the following table. ***(15pts)***

	MIPS Instruction	MACHINE CODE FORMAT
1.	`lui $t1,0x8C80`	00111100000010011000110010000000
2.	`addi $t0,$s0,-0x173E`	00100010000010000111010000011000010
3.	`or $18,$17,$0 = or $s2,$s1,$zero`	00000010001000001001000000100101
4.	`J L1 #L1 at address 0x7CAF83B4`	00001011001010111110000011101101

Q2-(25pts)

Suppose A,B and C are 32-bits signed integer local variables. The operation A+B will be executed in another procedure which is called "sum_procedure", and the result value will be returned to the main procedure and stored into variable C. Explain the required steps to translate this operation to MIPS assembly program.

To describe the above information by C/C++ Code segment

```
main()
{
int A,B,C;
C=sum_procedure(A,B);
}

int sum_procedure ( int x, int y)
{
int z;
z=x+y;
return(z);
}
```

∴ The required steps to translate this operation to MIPS assembly program as following:

1. Assign $a0 and $a1 to the arguments x and y respectively.
2. The instruction "jal sum_procedure" will store the PC+4 into $ra and jump to sum_procedure.
3. Acquire storage resources needed for procedure for example assign $t0 to variable z.
4. Perform the summation operation.
5. Place the result (z) into $V0.
6. Return to the caller by placing PC content with $ra register (jr $ra), and store $V0 into the register which assigned to variable C.

Q3-(25pts)

Consider a vector A : A is an 8-bits unsigned integer vector with four elements. Write a MIPS assembly program to calculate the summation of the last three elements and store the result into A[0]. *Assume vector A base address is corresponded to $t0.* ***(Hint: It can be done with six instructions.)***

Because A is 8-bit unsigned vector ---> we use lbu to load data into registers

```
lbu $s1,1($t0)  # Load the second element A[1] into $s1
lbu $s2,2($t0)  # Load the third element A[2] into $s2
lbu $s3,3($t0)  # Load the fourth element A[3] into $s3
add $t1,$s1,$s2 # $t1 = summation of A[1]+A[2]
add $t2,$t1,$s3 # $t2 = summation of A[1]+A[2]+A[3]
sb $t2, 0($t0)    # Store the result ($t2) into A[0]
```

Q4-(25pts)

Draw a 2 bit ALU circuit according to the following operation code written in the table below.

Operation	ALU operation code				
	S_4	S_3	S_2	S_1	S_0
A OR B	×	×	×	0	0
A AND B	×	×	×	0	1
A XOR B	×	1	0	1	0
A XNOR B	×	1	1	1	0
ADD(A+B)	0	0	0	1	1
SUB(A - B)	1	0	0	1	1